Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A computer implemented method for a pre-deployment analysis of a plurality of software components of an application prior to deployment of the application, comprising:

including, in an installation package for the application, a data structure that provides, for each of the plurality of software components from the application, a software component deployment dependency data, an indication of necessary software components for an operation of each of the plurality of software components being installed, and an indication of incompatibility with a previously installed software component; and

loading the installation package into a memory connected to a computer; and,

using the computer connected to the storage and a program installed in a memory of the computer so configured by the installation package, performing the steps of:

determining a first plurality of software components previously installed on a system;

determining a second plurality of software components to be installed on the system;

accessing a third plurality of software component deployment dependency data;

determining a fourth plurality of software components suitable for parallel installation;

determining an order in which the fourth plurality of software components can be grouped for a fifth plurality of parallel installations;

accessing a sixth plurality of metadata from the data structure regarding the second plurality of software components to be installed and accessing a seventh plurality of metadata regarding the first plurality of software components previously installed; and

analyzing the sixth plurality of metadata to determine an eighth plurality of potential conflicts between the second plurality of software components to be installed and the first plurality of software components previously installed on the system;

wherein the pre-deployment analysis allows the second plurality of software components to be installed in parallel and in a sequence of groups; and

wherein an installation time for the application is reduced.

2. (Canceled)

3. (Previously presented) The computer implemented method of claim 1, further comprising updating the data structure with an identity of a ninth plurality of recently installed software components.

4. (Previously presented) The computer implemented method of claim 1, further comprising providing a user with a plurality of options for the eighth plurality of potential conflicts.

5. (Previously presented) The computer implemented method of claim 4, wherein a first option includes aborting an installation.

6. (Previously presented) The computer implemented method of claim 4, wherein a second option includes continuing an installation.

7. (Previously presented) The computer implemented method of claim 6, further including, upon the exercise of the second option, recording an entry in a log indicative of a conflict and of a continuation of installation.

8. (Previously presented) The computer implemented method of claim 1, further comprising:

initiating a removal of a software component from the system; and

identifying a tenth plurality of remaining software components which depend on the software component to be removed.

9. (Previously presented) The computer implemented method of claim 8, further comprising providing a user with a plurality of options if the tenth plurality of dependent remaining software components are identified.

10. (Previously presented) The computer implemented method of claim 9, wherein a first option includes aborting a removal.

11. (Previously presented) The computer implemented method of claim 9, wherein a second option includes continuing a removal.

12. (Previously presented) The computer implemented method of claim 8, further comprising:

identifying a first software component previously installed on the system which is dependent upon a removed software component; and

determining an identity of a second software component upon which the first software component depends.

13. (Previously presented) The computer implemented method of claim 12, further comprising:

installing the second software component upon which the first software component depends; and

creating a dependency link between the first software component and the second software component.

14-40. (Canceled)

41. (Currently amended) A computer implemented method of using a semantic model to increase the efficiency of deployment of an application to a target by maximizing parallel installation of application software components, the computer implemented method comprising:

accessing the semantic model to obtain a dependency information about ~~the application~~ software components of an application;

including a semantic model in an installation package of the application;

responsive to loading the installation package into a memory connected to a computer, using the computer so configured by the installation package to perform steps comprising:

storing a first record of each of a plurality of the software components that is to be deployed in a read file;

storing a second record of each of a plurality of previously installed software components in a registry file;

when the read file is available to deploy, examining the registry file and accessing the semantic model to obtain a plurality of dependency information indicating a plurality of relationships among the plurality of the ~~application~~ software components to be installed in the target and among a plurality of previously installed software components;

using the dependency information to group the plurality of the ~~application~~ software components into sets of software components with like dependency levels, wherein a first set of software components from amongst the sets of software components has no dependencies, a second set of software components from amongst the sets of

software components has dependencies only on the first set of software components, and a third set of software components from amongst the sets of software components has dependencies only on the first and second sets of software components;

installing the first set of software components in parallel;

responsive to completing installation of the first set of software components, installing the second set of software components in parallel;

responsive to completing installation of the second set of software components, installing the third set of software components in parallel;

when a component is installed, updating the registry file;

when a conflict is identified, taking an appropriate action; and

displaying a progress report by labeling the plurality of the software components in the semantic model in a selected level of granularity.